



TITLE:

再帰的解析学における計算量 (証明論と複雑性)

AUTHOR(S):

河村, 彰星

CITATION:

河村, 彰星. 再帰的解析学における計算量 (証明論と複雑性). 数理解析研究所講究録 2013, 1832: 33-38

ISSUE DATE:

2013-04

URL:

<http://hdl.handle.net/2433/194856>

RIGHT:

再帰的解析学における計算量

河村彰星 (東大・情報理工)

解析学の様々な対象について再帰(計算可能性)理論の観点から調べるのが再帰的解析学(Recursive/Computable Analysis)である[Wei, BHW]. これを精密化して時間等の資源を限った計算を考えることは葛とフリードマン[KF]に始まる. 離散的な問題の困難さを理解する尺度として確立した多項式時間 P, 非決定的多項式時間 NP, 多項式空間 PSPACE 等の計算量級に関する理論を, 実数など連続量の絡む問題に応用するのである. 本稿ではこの分野の典型的な結果の例と最近の展開を最小限の定義とともに紹介する.

多項式時間計算可能実函数

文字列函数 $\varphi: \{0, 1\}^* \rightarrow \{0, 1\}^*$ が実数の名であるとは, 任意の $n \in \mathbb{N}$ について $\varphi(0^n)$ が x から距離 2^{-n} 未満の有理数(を文字列で表示したもの)であることをいう.

実函数の多項式時間計算可能性等を定義するにはチューリング機械がそのような名を神託として読み書きすると考える. 即ち

定義 1 実函数 $f: [0, 1] \rightarrow \mathbb{R}$ を神託つきチューリング機械 M が計算するとは, 任意の $t \in [0, 1]$ の任意の名 φ について M^φ が $f(t)$ の或る名であることをいう(図 1).

但し機械 M に神託 φ と文字列 u とを与えたとき出力される文字列を $M^\varphi(u)$ で表している^{*1}.

この機械 M が多項式時間であるとは入力文字列の長さ(図 1 では m)に対してである^{*2}. 特に機械は値 $f(t)$ の 2^{-m} 近似を出力するまでに t の 2^{-n} 近似(或る多項式 p につ

^{*1} 機械が停止しないと $M^\varphi(u)$ が定まらないが, ここでは多項式時間に限られた計算のみを考えるため必ず停止するとしてよい. また神託 φ の返す答が質問に比べて非常に長いと機械がこれを制限時間内に読めないから, 厳密には「任意の」名 φ と言っては拙いのだが, 実数 t の 2^{-n} 近似 $\varphi(0^n)$ は $O(n)$ 桁程度で表されるので, 差当り φ はそのような冗長でない名に限ると考える(より自然な解決は[KawC]のように制限時間の測り方を緩めることである).

^{*2} ここでは機械の動作時間は m から決り神託即ち実数 t に依らないとした. $t \in [0, 1]$ の代りに一般の対象を考える際には後述のようにこの点を修正する.

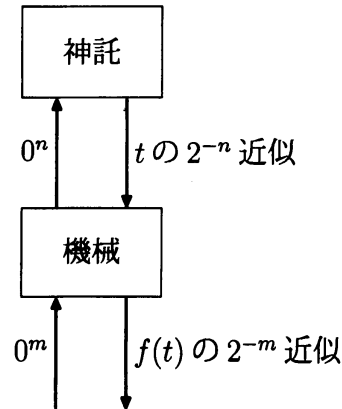


図1 実函数 f を計算する神託チューリング機械

いて $n \leq p(m)$ を知るのみであるから, f は連続でなければならない.

例えば正弦函数 \sin (を区間 $[0, 1]$ へ制限したもの) は多項式時間可計算である. 何となれば $[0, 1]$ 上において級数

$$\sin t = t - \frac{t^3}{3!} + \frac{t^5}{5!} - \frac{t^7}{7!} + \cdots$$

は十分速く収斂するため, その和を精度 2^{-m} で近似するには初めの $m+1$ 項の和を精度 2^{-m-1} で求めれば足るからである.

実函数の計算量の保存

この概念を使って実函数に対する操作の複雑さについて或る程度のことを述べることができる. 例えば次のような定理が成立つ(2変数函数の計算は神託を二つ取る機械により同様に定義する).

定理 2 [Fri] 次は同値.

- $P = NP$.
- 任意の多項式時間計算可能函数 $g: [0, 1]^2 \rightarrow \mathbb{R}$ について, 次の函数 $h = \text{MAX } g: [0, 1] \rightarrow \mathbb{R}$ は多項式時間計算可能である.

$$h(t) = \max_{y \in [0, 1]} g(t, y) \quad (t \in [0, 1])$$

定理 3 [Kol, Kaw, KORZ] 次は同値.

表 1 g が多項式時間計算可能であるときの解 h の計算量(各結果の出典は[Kaw]または[KORZ]参照).

仮定	上界	下界
なし	——	(複数の解 h が) みな計算不能であり得る
h は唯一の解	計算可能	任意に多くの時間・空間を要し得る
g が[Ko3]の意味で 局所リプシッツ連続	指数空間計算可能	EXPSPACE 困難であり得る
g はリプシッツ連続	多項式空間計算可能	PSPACE 困難であり得る
g は連続可微分	多項式空間計算可能	PSPACE 困難であり得る (定理 3)
g は k 回連続可微分 ($k \geq 2$)	多項式空間計算可能	CH 困難たり得る
g は解析的	多項式時間計算可能	——

- $P = PSPACE$.
- 任意の連続微分可能な多項式時間計算可能函数 $g: [0, 1] \times [-1, 1] \rightarrow \mathbb{R}$ について, 次で定まる函数 $h = ODE\ g: [0, 1] \rightarrow \mathbb{R}$ は(もし $[0, 1]$ 上で $[-1, 1]$ 内の値を取り続ければ)多項式時間計算可能である.

$$h(0) = 0 \qquad h(t) = g(t, h(t)) \qquad (t \in [0, 1])$$

なお定理 3 の前提においては g に対する連続微分可能な条件から解 h の一意性が保せられているのであるが, この条件を変えると h の計算量が表 1 のようになることが判っている. この〇〇困難は[Ko3]のように適当に定義したもので, 定理 3 は h の PSPACE 困難性を述べている(表 1 の該当行)のとはほぼ同じことになる.

以上の実函数の多項式時間計算可能性の定義は或る程度拡張でき, 色々な対象の計算量が調べられている. 本稿では扱わないが実函数でなく実数集合に対する多項式時間計算可能性(幾つか定義がある)も調べられている. 葛[Ko2, Ko4]に詳しい.

より一般の演算子の計算量

定理 2, 3 は MAX や ODE が実函数の多項式時間計算可能性を保つか否かを, 通常の計算量級に関連づける形で述べているが, 少し定義の手間を掛けて二階計算量の議論を応用することで, MAX や ODE そのものの複雑さについての「一様」「構成的」な主張に強める

こともできる[KawC]. その方法の概略を述べる.

定義 1 において実数の文字列関数 $\varphi: \Sigma^* \rightarrow \Sigma^*$ への符号化を通して実関数の多項式時間計算可能性を定義したように, 実関数を符号化することで実関数をとる演算子の計算量を定義したいのであるが, 実はそのように計算の対象を広げようとする, 機械の動作時間が神託によらず入力文字列のみの多項式で抑えられねばならないという制約が不都合になる. そこで φ の「長さ」を定義し, それに対する「多項式」時間であることを定義する.

まず計算対象を符号化するのに使う関数 $\varphi: \Sigma^* \rightarrow \Sigma^*$ を, **長さ単調**であるもの, 即ち $|u| \leq |v|$ なる任意の文字列 u, v に対して $|\varphi(u)| \leq |\varphi(v)|$ が成立つものに限ることにする. この制限は符号化を行う上では本質的な不便を生ずるものではない. 長さ単調な関数 φ は特に同じ長さの文字列を同じ長さの文字列へ移すから, φ の**長さ** $|\varphi|: \mathbb{N} \rightarrow \mathbb{N}$ を $|\varphi|(|u|) = |\varphi(u)|$ により定義できる.

この $|\varphi|$ は「長さ」と言っても \mathbb{N} から \mathbb{N} への関数であるから, それに関する多項式というものを定義する必要がある. そこで(一型変数 L と零型変数 n の)**二階多項式**を次の如く帰納的に定義する. 正整数は二階多項式である. 変数 n は二階多項式である. もし P と Q とが二階多項式ならば, $P+Q$ や $P \cdot Q$ や $L(P)$ もまた然り. この定義はカプロンとクック[KapC]が多項式時間汎関数[Meh]を特徴づけた方法に倣ったものである. 例えば

$$L(L(n \cdot n)) + L(L(n) \cdot L(n)) + L(n) + 4$$

は二階多項式である. 二階多項式 P は関数 $L: \mathbb{N} \rightarrow \mathbb{N}$ を関数 $P(L): \mathbb{N} \rightarrow \mathbb{N}$ へ移す関数を定める. 例えば P が上の二階多項式であって $L(x) = x^2$ ならば, $P(L)$ は

$$P(L)(x) = ((x \cdot x)^2)^2 + (x^2 \cdot x^2)^2 + x^2 + 4 = 2 \cdot x^8 + x^2 + 4$$

という関数である. 機械 M が**多項式時間**であるとは, 二階多項式 P が存在し, M に如何なる長さ単調な神託 φ と如何なる文字列入力 $u \in \Sigma^*$ とを与えても, 遷移 $P(|\varphi|)(|u|)$ 回以内で停止することとする.

こうして定義される多項式時間計算可能性は, 通常が多項式時間を保つ自然な二階への拡張になっており, 良い性質をもつ安定した概念となる. 非決定性計算や空間計算量についてもうまく拡張でき, **P**, **NP**, **PSPACE** などの級を定義できる(二階の計算量であることを示すため太字で書くことにする). また問題間の帰着(計算可能性においては既にワイラオホ帰着として知られていたもの)も適切に定義できる. これを用いて定理 2, 3 は, 実関数の適当な符号化の下で

定理 2' **MAX** は **NP** 完全である.

定理 3' ODE は PSPACE 完全である.

という形に強めることができる(もとの定理 2, 3 はその系となる). これはただ主張が強くなったのみならず, 与えられた関数 g から h を計算するという演算そのものの複雑さ(数値算法で解くべき問題の困難さ)をより直接に捉えている. 詳細は[KawC]参照.

今後の研究課題

このように従来計算可能性の論じられてきた解析学の諸問題を, 多項式時間を始めとする計算量の世界で扱うための枠組は整ってきたが, 対象とし得る空間及び計算量の種類という点でなお理論的整備の余地があり, 具体的な問題の計算量についても既知の結果は少い. また次のような他分野との接点にも研究課題がある.

数値算法との関係 再帰的解析学の計算モデルは必ず所定の精度で解を求めるという厳しいものであるが, 計算機の高速化によって近年では数値計算の分野でも精度保証が見直され, 実用を目指した実装も進められている[Mül]. そのような数値算法の可能性と限界を明らかにするため, 応用上重要な多くの数値計算問題の困難さを調べるのが課題となる. また実際の数値算法では計算量をより精密に扱う必要があるが, 例えば二階多項式には通常多項式の「次数」のようなものがないため, 色々な算法を組合せて使う中で如何なる指標が実際の計算効率に与っているのか検討を要する.

形式体系との関係 計算可能性の水準では, 様々な対象の存在を示すに要する公理について「逆数学」の諸結果がある. 例えば定理 3 より弱い条件で微分方程式の解の存在を述べるペアノ定理は弱ケーニヒの補題に同値である[Sim]. これは同条件下で計算不能な解(表 1 最上行)が生ずる「理由」を或る程度説明している. 同様のことが P や PSPACE に相当する弱い体系で考えられるだろうか. これは定理 3, 表 1 のような現象を良く理解することにも通ずる自然な興味と思われるが(「有界逆数学」[CN])現在の所ほぼ未着手のようである.

文献

- [BHW] V. Brattka, P. Hertling and K. Weihrauch. A tutorial on computable analysis. In *New Computational Paradigms: Changing Conceptions of What is Computable*, S.B. Cooper, B. Löwe, and A. Sorbi, Eds. Springer, 2008, pp. 425–491.
- [CN] S. Cook and P. Nguyen. *Logical foundations of proof complexity*. Cambridge University Press, 2010.

- [Fri] H. Friedman. The computational complexity of maximization and integration. *Advances in Mathematics* 53, 80–98, 1984.
- [KapC] B. M. Kapron and S. A. Cook. A new characterization of type-2 feasibility. *SIAM Journal on Computing* 25, 117–132, 1996.
- [Kaw] A. Kawamura. Lipschitz continuous ordinary differential equations are polynomial-space complete. *Computational Complexity* 19, 305–332, 2010.
- [KawC] A. Kawamura and S. Cook. Complexity theory for operators in analysis. *ACM Transactions on Computation Theory* 4, Article 5, 2012.
- [KF] K. Ko and H. Friedman. Computational complexity of real functions. *Theoretical Computer Science* 20, 323–352, 1982.
- [KORZ] A. Kawamura, H. Ota, C. Rösnick and M. Ziegler. Computational complexity of smooth differential equations. In *Proc. 37th International Symposium on Mathematical Foundations of Computer Science (MFCS 2012)*, LNCS 7464, 578–589, 2012.
- [Ko1] K. Ko. On the computational complexity of ordinary differential equations. *Information and Control* 58, 157–194, 1983.
- [Ko2] K. Ko. *Computational Complexity of Real Functions*. Birkhäuser Boston, 1991.
- [Ko3] K. Ko. On the computational complexity of integral equations. *Annals of Pure and Applied Logic* 58, 201–228, 1992.
- [Ko4] K. Ko. Polynomial-time computability in analysis. In *Handbook of Recursive Mathematics* Vol. 2, Yu. L. Ershov et al., Eds., 1271–1317, 1998.
- [Meh] K. Mehlhorn. Polynomial and abstract subrecursive classes. *Journal of Computer and System Sciences* 12, 147–178, 1976.
- [Mül] N. Th. Müller. The iRRAM: Exact arithmetic in C++. LNCS 2064, 222–252, 2001.
- [Sim] S. G. Simpson. Which set existence axioms are needed to prove the Cauchy/Peano Theorem for ordinary differential equations? *Journal of Symbolic Logic* 49, 783–802, 1984.
- [Wei] K. Weihrauch, *Computable Analysis*. Springer, 2000.